Amendments to the Claims

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Currently Amended) A data access method, comprising a data reading procedure to read a data from a data storage zone wherein said data is stored in a bit range of said data storage zone covering at least one storage units, each storage unit of said data storage zone consisting of m bits, and said bit range consists of n bits from a starting bit address (a) to an end bit address (b), and said data reading procedure comprising steps of:

i) performing a first operation of said starting bit address (a) to obtain a first shift S1;

ii) performing a second operation of said starting bit address (a) to obtain a second shift S2;

iii) performing a first shift operation of said data with said first shift S1 to obtain a first shifted data unit;

iv) performing a second shift operation of said data with said second shift S2 to obtain a second shifted data unit;

v) synthesizing said first and said second shifted data units to obtain a read data unit; and

vi) repeating at least one of said steps iii), iv) and v) when n is greater than m;

wherein said first and said second operations are performed by the following formulae:

S1 = mod [a, m]; and

S2 = m – mod [a, m] = m – S1,

where mod [a, m] is the remainder on division of a by m.

2. (Cancelled).

3. (Cancelled)

4. (Currently Amended) The data access method according to claim ~~3~~ 1 wherein said first shift operation is performed by shifting a first portion of said data stored in a first storage unit of said data storage zone toward one of the higher bit direction and the lower bit direction, and said second shift operation is performed by shifting a second portion of said data stored in a second storage unit of said data storage zone toward the other of the higher bit direction and the lower bit direction.

5. (Previously Presented) The data access method according to claim 4 wherein said second storage unit is immediately adjacent to said first storage unit in said data storage zone.

6. (Previously Presented) The data access method according to claim 5 wherein in said step (vi), only said step (iii) is repeated for shifting an end data unit comprising said end data bit address (b) with said first shift S1 to obtain a last shifted data unit.

7. (Original) The data access method according to claim 6 further comprising a step of masking said last shifted data unit with a mask data MD for clearing bits excluded from said bit range, where MD = 0xFF >> (m - (b-a+1)), the expression "0xFF" indicates an 8-bit hexadecimal mask data and the 8 bits are all "1", and the expression "X >> Y" indicates the rightward shift of the data X by Y bits.

8. (Original) The data access method according to claim 1 wherein said first and said second shifted data units are synthesized via an OR gate operation.

9. (Currently Amended) A data access method, comprising a data writing procedure to write a data into a data storage zone, said data storage zone storing said data in a bit range covering at least one storage unit, each storage unit of said data storage zone consisting of m bits, said

bit range consisting of n bits from a starting bit address (a) to an end bit address (b), and said data writing procedure comprising steps of:

i) performing a first operation of said starting bit address (a) to obtain a first shift S3;

ii) performing a second operation of said starting bit address (a) to obtain a second shift S4;

iii) performing a first shift operation of said data with said second shift S4 to obtain a first shifted data unit;

iv) performing a second shift operation of said data with said first shift S3 to obtain a second shifted data unit; and

v) synthesizing said first and said second shifted data units to obtain a written data unit,

wherein at least one of said steps iii), iv) and v) is preformed more than once when n is greater than m; and

wherein said first and said second operations are performed by the following formulae:

S3 = mod [a, m]; and

S4 = m – mod [a, m] = m – S3,

where mod [a, m] is the remainder on division of a by m.

10. (Cancelled)

11. (Cancelled)

12. (Currently Amended) The data access method according to claim ~~11~~ 9 wherein said first shift operation is performed by shifting a first data unit of said data to be written toward one of the higher bit direction and the lower bit direction to obtain said first shifted data unit, and said second shift operation is performed by shifting a second data unit of said data to be written toward the other of the higher bit direction and the lower bit direction to obtain said second shifted data unit.

13. (Original) The data access method according to claim 12 wherein said second data unit is immediately adjacent to said first data unit in said data storage zone.

14. (Previously Presented) The data access method according to claim 13 further comprising steps of:

determining whether said second data unit is the last data unit of said data to be written, wherein said first shifted data unit and said second shifted data unit are synthesized to obtain an end written data unit when said second data unit is the last data unit of said data to be written; and

performing a masking procedure with a mask data MD3 for clearing bits of an end storage unit of said storage zone for storing said end written data unit when said second data unit is the last data unit of said data to be written, where MD3 = 0xFF << (mod [b, m] +1), mod [b, m] is the remainder on division of b by m, the expression "0xFF" indicates an 8-bit hexadecimal mask data and the 8 bits are all "1", and the expression "X << Y" indicates the leftward shift of the data X by Y bits.

15. (Original) The data access method according to claim 13 wherein said first and said second shift operations are further performed on subsequent data units until the last data unit of said data to be written has been shifted.

16. (Previously Presented) The data access method according to claim 12 further comprising before said step (iii) steps of:

determining whether said first data unit is the starting data unit of said data to be written;

performing a starting shifting operation of said first data unit with said first shift S3 to obtain a starting shifted data unit when said first data unit is the starting data unit of said data to be written; and

performing a masking procedure with a mask data MD2 for clearing bits of a starting storage unit of said storage zone for storing said starting shifted data unit,

where MD2 = ~(0xFF << S3), the expression "0xFF" indicates an 8-bit hexadecimal mask data and the 8 bits are all "1", the expression "X << Y" indicates the leftward shift of the data X by Y bits, and the expression "~Z" indicates the reverse logic operation of data Z.

17. (Original) The data access method according to claim 9 wherein said first and said second shifted data units are synthesized via an OR gate operation.

18. (Previously Presented) A data access method, comprising a data writing procedure to write a data into a data storage zone, said data storage zone storing data in a bit range covering at least one storage unit, each storage unit of said data storage zone consisting of m bits, said bit range consisting of n bits from a starting bit address (a) to an end bit address (b), and said data writing procedure comprising steps of:

performing a first clear and writing procedure of said data to be written when n is no greater than m, said first clear and writing procedure comprising a step of masking said bit range with a first mask data MD1 = ~((0xFF >> ((m-1) – b + a)) << mod [a, m]); and

performing a second clear and writing procedure and a third clear and writing procedure of said data to be written when n is greater than m, said second clear and writing procedure comprising a step of masking a starting storage unit with a second mask data MD2 = ~(0xFF << mod [a, m]), and said third clear and writing procedure comprising a step of masking an end storage unit with a third mask data MD3 = 0xFF << (mod [b, m] +1);

where the expression "0xFF" indicates a hexadecimal mask data, the expression "X >> Y" indicates the rightward shift of the data X by Y bits, the expression "X << Y" indicates the leftward shift of the data X by Y bits, the expression "~Z" indicates the reverse logic operation of data Z, the expression "X & Y" indicates AND gate operation of data X and Y, the expression "mod [a, m]" indicates the remainder on division of a by m, and the expression "mod [b, m]" indicates the remainder on division of b by m.

19. (Original) The data access method according to claim 18 wherein said data writing procedure is performed as little endian.

20. (Original) The data access method according to claim 18 wherein said data writing procedure is performed as big endian.

21. (Previously Presented) The data access method according to claim 18 wherein when n is greater than m, the starting data unit of said data is shifted by a shift S3 and then written into said starting storage unit of said data storage zone in said second clear and writing procedure, where S3 = mod [a, m] that is the remainder on division of a by m.

22. (Previously Presented) The data access method according to claim 18 wherein when n is greater than m, the last second data unit and the last data unit of said data are shifted by a first shift S3 and a second shift S4, respectively, and the differentially shifted data are synthesized and then written into said end storage unit in said third clear and writing procedure, where S3 = mod [a, m] that is the remainder on division of a by m, and S4 = m – S3.